GitHub使用

1. 初始化一个Git仓库，使用git init命令。
2. 创建一个版本库非常简单，首先，选择一个合适的地方，创建一个空目录：
3. $ mkdir learngit
4. $ cd learngit
5. $ pwd
6. C:/Users/lee/learngit
7. pwd命令用于显示当前目录。在我的电脑上，这个仓库位于C:/Users/lee/learngit
8. 第二步，通过git init命令把这个目录变成Git可以管理的仓库：
9. $ git init
10. 瞬间Git就把仓库建好了，而且告诉你是一个空的仓库（empty Git repository），细心的读者可以发现当前目录下多了一个.git的目录（隐藏文件):
11. 这个目录是Git来跟踪管理版本库的，没事千万不要手动修改这个目录里面的文件，不然改乱了，就把Git仓库给破坏了。
12. git status

查看本地修改后的工程与上一版工程之间的更改状态信息；

1. git diff “文件名”

虽然Git告诉我们readme.txt被修改了，但如果能看看具体修改了什么内容，自然是很好的。比如你休假两周从国外回来，第一天上班时，已经记不清上次怎么修改的readme.txt，所以，需要用这个命令。

1. git add code\_file\_name1 code\_file\_name2 … code\_file\_namen

添加的代码文件到git仓库

1. git commit -m "任意起名"

提交更改后的工程到git仓库（git commit 主要是将暂存区里的改动给提交到本地的版本库。）

1. git push

在GitHub账户上更新工程

1. git reset - -hard commit\_id 时光机穿梭

HEAD指向的版本就是当前版本，因此，Git允许我们在版本的历史之间穿梭，使用命令git reset - -hard commit\_id。 在Git中，用HEAD表示当前版本，，上一个版本就是HEAD^，上上一个版本就是HEAD^^，当然往上100个版本写100个^比较容易数不过来，所以写成HEAD~100

1. 穿梭前，用git log可以查看提交历史，以便确定要回退到哪个版本。
2. 要重返未来，用git reflog查看命令历史，以便确定要回到未来的哪个版本。
3. 提交后，用git diff HEAD -- readme.txt命令可以查看工作区和版本库里面最新版本的区别：